REMARKS

Claims 1, 18, 25, and 41 have been amended to clarify the invention. Claims 1-8 and 10-41 remain pending.

The Examiner rejected claims 1-8 and 10-41 under 35 U.S.C. §103(a) as being unpatentable over Li (U.S. Patent 5,643,125) and further in view of Dreszer (U.S. Patent 6,442,661). The Examiner's rejections are respectfully traversed as follows.

Claim 1 is directed towards a "method for assigning traffic buckets to a cache system." Claim also requires that "a) when a new cache system starts up in a cache cluster having a plurality of cache systems among which a plurality of total buckets are to be allocated, determining a full bucket allocation for the new cache system; b) periodically determining a load of the new cache system; c) each time it is periodically determined that the new cache system is underloaded, slowly assigning a portion of the full bucket allocation when buckets have not been previously shed from the new cache system and slowly assigning a portion of previously shed buckets to the new cache system when buckets have been previously shed from the new cache system, unless the cache cluster is operating at a maximum load and d) each time it is periodically determined that the new cache system is overerloaded, shedding a portion of the buckets previously assigned to the new cache system." Claim 1 also requires "wherein each bucket portion corresponds to a portion of the total traffic being handled by the cache cluster." Independent claims 18, 25, and 41 recite mechanisms for periodically determining a load of a new cache system that starts up in a cache cluster having a plurality of cache systems among which a plurality of total buckets are to be allocated and then slowly assigning and shedding buckets to and from the new cache system based on each of the periodic load determinations for the new cache system, wherein each bucket portion corresponds to a portion of the total traffic being handled by the cache cluster.

The present invention allows efficient distribution of traffic buckets among a plurality of cache systems of a cache cluster system by periodically monitoring the load of a new cache system when it starts up in a cache cluster having a plurality of cache systems. Bucket *portions* are *slowly* assigned to the new cache system. Based on this periodic monitoring of the new cache system's load, traffic bucket portions of the full allocation are slowly assigned to this new cache system when buckets have not been previously shed, which allows the new cache system to not be quickly overwhelmed by too much traffic. By applying this technique to each new cache system as it enters the cache cluster, traffic buckets can be periodically distributed among the cache systems of the cache cluster while managing the traffic load of individual cache systems so they don't remain overloaded.

Conventionally, traffic buckets were evenly allocated between the available cache systems of a cache cluster and the full allocation is immediately assigned to each cache system. For various reasons, a particular cache system may be overloaded with traffic under this simple distribution scheme. For instance, a particular cache system may be unable to handle as much traffic as another cache system. See Background Section, Page 4, Lines 16-22.

The primary reference Li describes a parallel database system for storing tables at multiple nodes. Li also describes techniques for storing tables at a node that is being added to the parallel database system in Col. 5, Line 38 through Col. 6, Line 52.

Even if one argues that distribution of tables among nodes is the same as distribution of traffic bucket portions to cache systems, Li fails to teach or suggest slow assignment of traffic bucket portions of the full bucket allocation for a new system when buckets have not been previously shed from such new system, in the manner claimed. That is, Li fails to teach *slowly* assigning a portion of the full bucket allocation in the manner claimed.

The Examiner argues that since Li teaches that "in the quiescent mode operation, load balancing is the primary goal", Li teaches periodic monitoring of the load of a new cache system. However, it is respectfully submitted that a load balancing technique is not necessarily periodic and can be a single operation where the load is monitored once when the new node starts up. The reference Li fails to teach any type of periodic monitoring of the load of a new cache system, and such a teaching cannot be implied since load balancing is not necessarily periodic. The Examiner assertion that periodic monitoring of the load of a new cache system is well known is challenged, and the Examiner is respectfully asked to provide a reference that teaches such feature.

Even if one argues that Li teaches periodic monitoring of a new cache system, it is respectfully submitted that Li fails to teach a slow assignment of portions of the full bucket allocation to a new cache system when buckets have not been previously shed from such new cache system, in the manner claimed.

The secondary reference Dreszer also suffers from the above described deficiencies. That is, Dreszer fails to teach periodic monitoring of the load of a new cache system and slow assignment of portions of the full bucket allocation to a new cache system when buckets have not been previously shed from such new cache system, in the manner claimed. In other words, Dreszer fails to teach two steps, (1) determining a full bucket allocation for the new cache system and (2) then slow assignment of portions of the full bucket allocation to a new cache system, in the manner claimed.

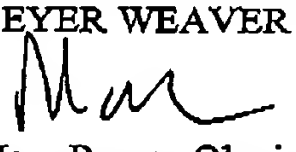
In general, Dreszer is directed towards managing a memory cache of a single system and not assigning buckets to a new cache system that is starting up in a cache system having multiple cache systems among which buckets are allocated, in the manner claimed. Although Dreszer teaches periodically trimming queue sizes in a single system, Dreszer merely teaches trimming the queue sizes based on each memory request. See Column 12, Lines 21-22: "Fig. 11 shows an example flow diagram of an embodiment of step 70 of FIG. 4 for adjusting the sizes queues 40 in relation to memory request/requirements" and Lines 62-66: "the selected size queue 40 includes a free segment 42 of sufficient size, wherein the segment 42 is allocated to satisfy the request (step 232)." In other words, the size queue's segment is allocated to based on the request's size requirement. That is, the size queue's segment allocation is not slowly assigned to the size queue in portions, in the manner claimed.

For the forgoing reasons, it is respectfully submitted that claims 1, 18, 25, and 41 are patentable over the cited references.

The Examiner's rejections of the dependent claims are also respectfully traversed. However, to expedite prosecution, all of these claims will not be argued separately. Claims 2-9, 11-17, 19-24, and 26-40 each depend directly from independent claims 1, 18, or 25 and, therefore, are respectfully submitted to be patentable over cited art for at least the reasons set forth above with respect to claims 1, 18, and 25. Further, the dependent claims require additional elements that when considered in context of the claimed inventions further patentably distinguish the invention from the cited art.

Applicant believes that all pending claims are allowable and respectfully requests a Notice of Allowance for this application from the Examiner. Should the Examiner believe that a telephone conference would expedite the prosecution of this application, the undersigned can be reached at the telephone number set out below.

Respectfully submitted,
BEYER WEAVER & THOMAS, LLP



Mary Ramos Olynick
Reg. 42,963

P.O. Box 778
Berkeley, CA 94704-0778
(510) 843-6200